

Cookies and Sessions

Maintaining State in HTTP

open.michigan

Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.
<http://creativecommons.org/licenses/by/3.0/>.

Copyright 2009, Charles Severance



University of Michigan School of
INFORMATION



High Level Summary

- The web is “stateless” - the browser does not maintain a connection to the server while you are looking at a page. You may never come back to the same server - or it may be a long time - or it may be one second later
- So we need a way for servers to know “which browser is this?”
 - In the browser state is stored in “Cookies”
 - In the server state is stored in “Sessions”

Welcome to SI539

Please Log In

Required Information

Enter your E-Mail:

Enter your Password:

If you have lost your password,
membership@si539.com to have

UNIVERSITY OF MICHIGAN WEBLOGIN

AUTHENTICATION REQUIRED::

You are connecting to a U-M website that requires authentication. Please enter your Login ID (username or Friend ID) and password to continue.

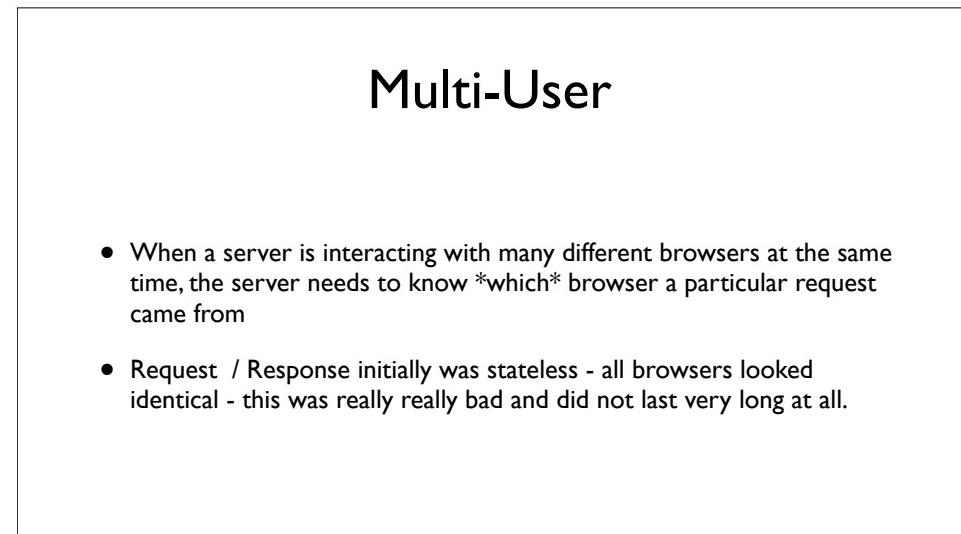
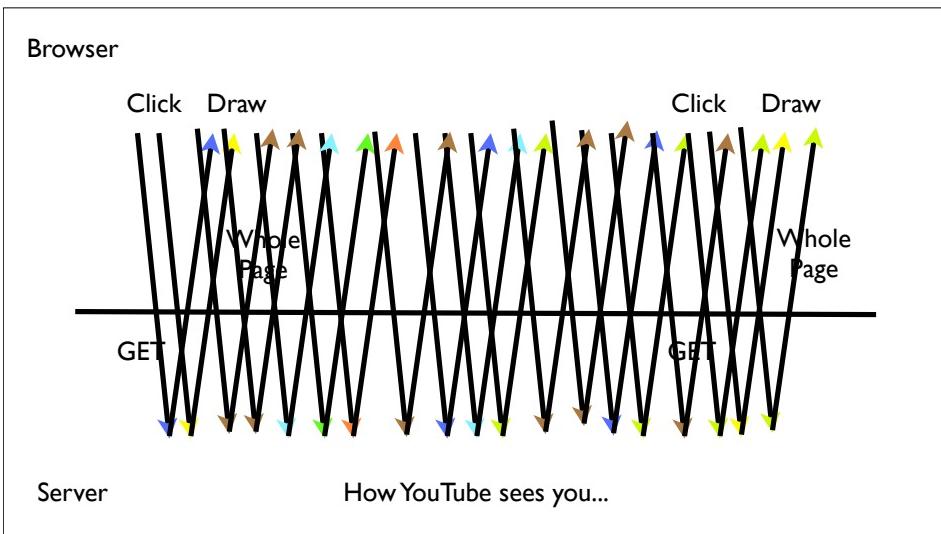
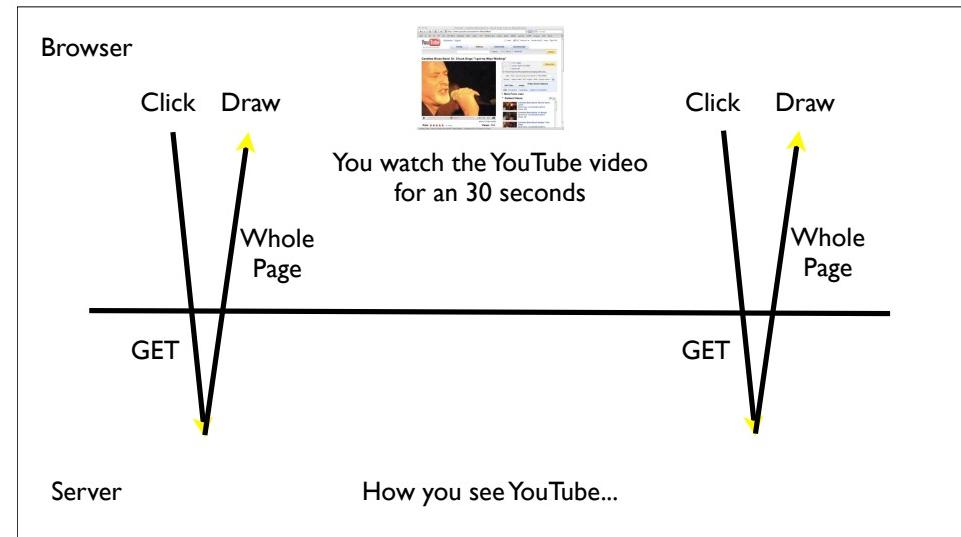
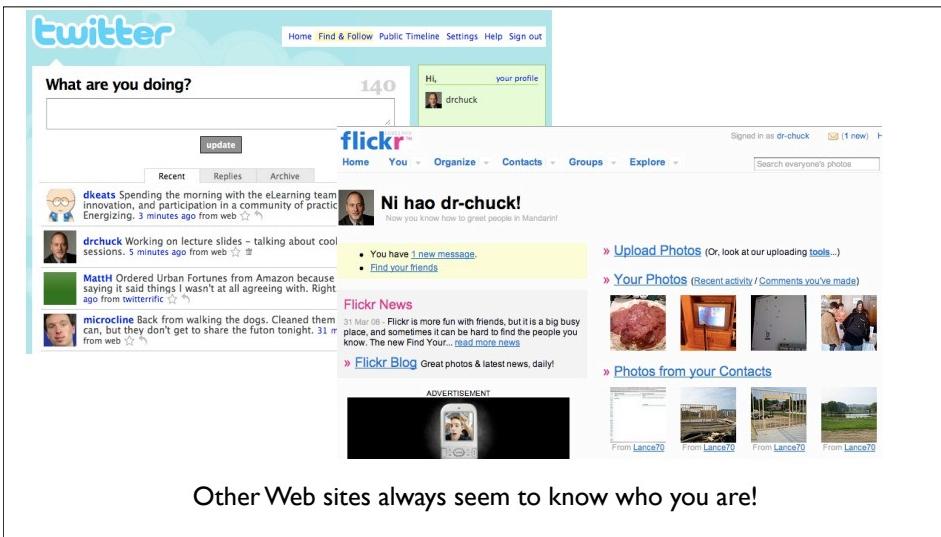
Need a Login ID?

If you don't have a Login ID, you can [create one now](#).

Login ID
Password
► MToken
[Forgot your password?](#)
[Login Help](#)

By using this service you agree to adhere to [U-M computing policies and guidelines](#).

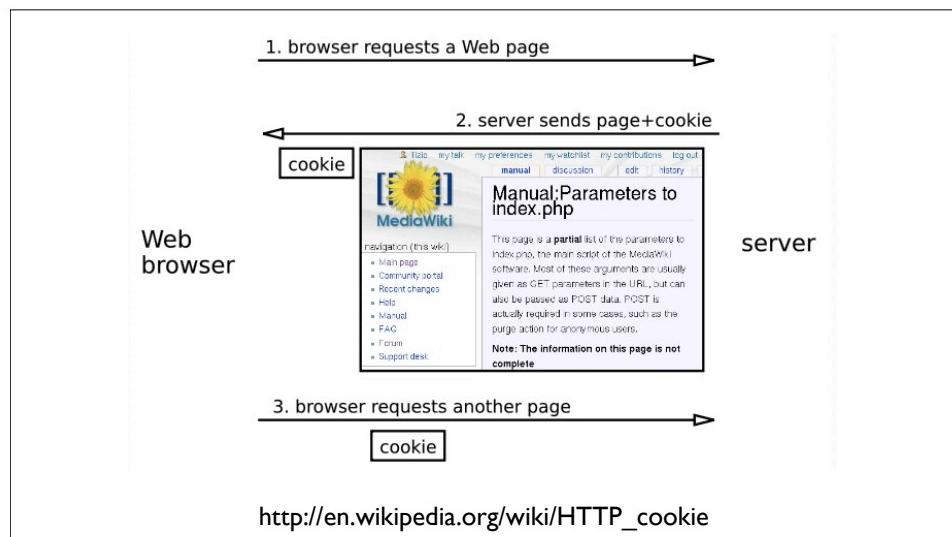
Some Web sites always seem to want to know who you are!



Web Cookies to the Rescue

Technically, cookies are arbitrary pieces of data chosen by the Web server and sent to the browser. The browser returns them unchanged to the server, introducing a state (memory of previous events) into otherwise stateless HTTP transactions. Without cookies, each retrieval of a Web page or component of a Web page is an isolated event, mostly unrelated to all other views of the pages of the same site.

http://en.wikipedia.org/wiki/HTTP_cookie



Cookies In the Browser

- Cookies are marked as to the web addresses they come from - the browser only sends back cookies that were originally set by the same web server
- Cookies have an expiration date - some last for years - others are short-term and go away as soon as the browser is closed

Playing with Cookies

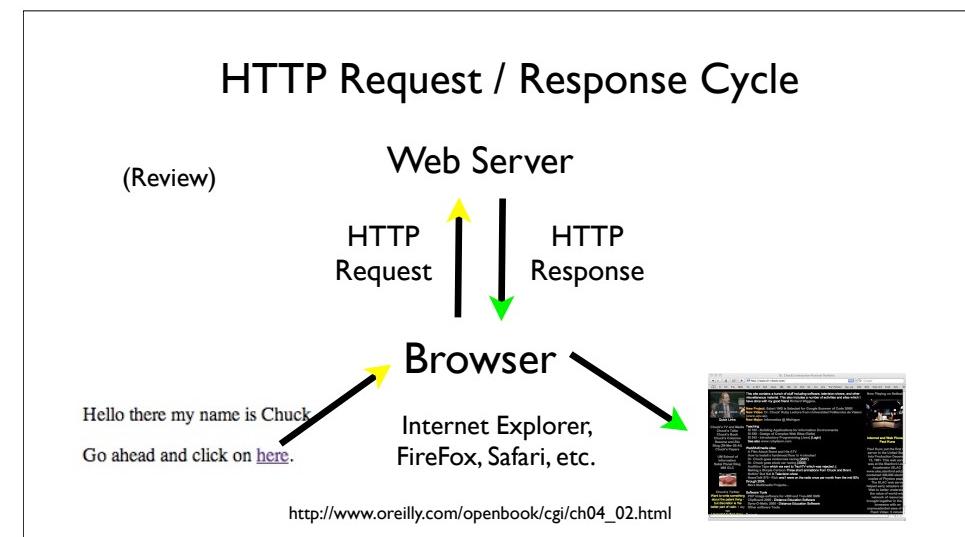
- Firefox Developer Plugin has a set of cookie features
- Other browsers have a way to view or change cookies

The screenshot shows a web browser window with the title 'Assignment 10 - SI539'. The main content area displays the 'Welcome to SI539' page, which includes a sidebar with links like 'About', 'Contact', 'Pictures', 'Membership', 'Chat', and 'Logout'. The main content area contains two quotes from Ghandi and a note about power. Below the quotes, there is a note: 'I hope to demonstrate that the real authority by a few, but by the acquisition of authority, when abused.' To the right of the browser window, a 'Cookie Information' extension is open, showing a list of cookies for the current page. One cookie is selected, showing details: NAME: __assn6_session_id, VALUE: f33e49cd915aad8b9a1f768b8081aa6, HOST: localhost, PATH: /, SECURE: No, EXPIRES: At End Of Session.

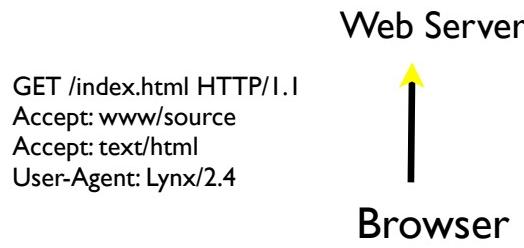
Cookies

- Identifying Individual Users
- The Web is “stateless”
- How do we make the web seem not to be stateless

Request Response Again!



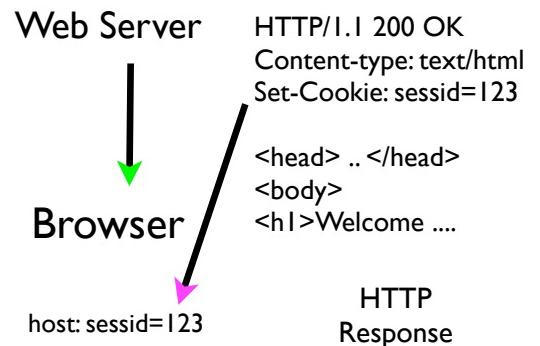
HTTP Request / Response Cycle



We do our initial GET to a server. The server checks to see if we have a cookie with a particular name set. Since this is our first interaction, we have not cookies set for this host.

http://www.oreilly.com/openbook/cgi/ch04_02.html

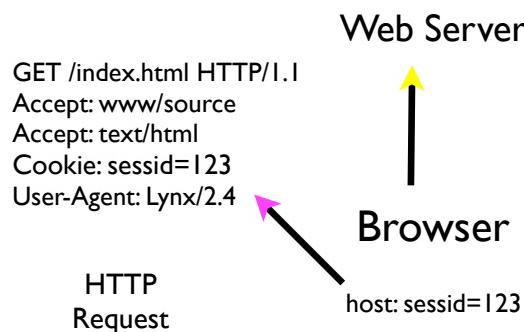
HTTP Request / Response Cycle



Along with the rest of the response, the server sets a cookie with some name (sessid) and sends it back along with the rest of the response.

http://www.oreilly.com/openbook/cgi/ch04_02.html

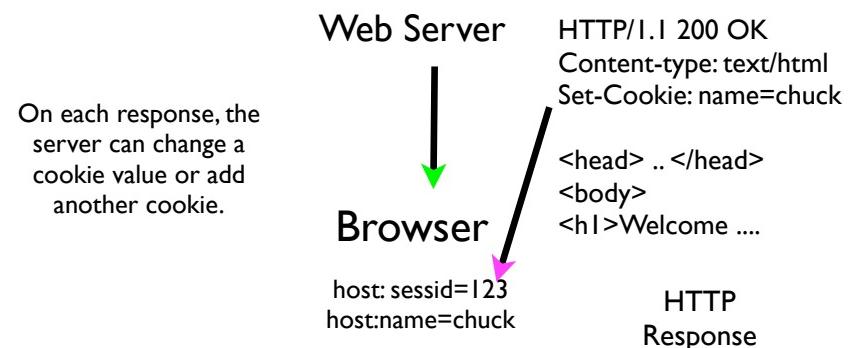
HTTP Request / Response Cycle



From that point forward, each time we send a GET or POST to the server, we include any cookies which were set by that host.

http://www.oreilly.com/openbook/cgi/ch04_02.html

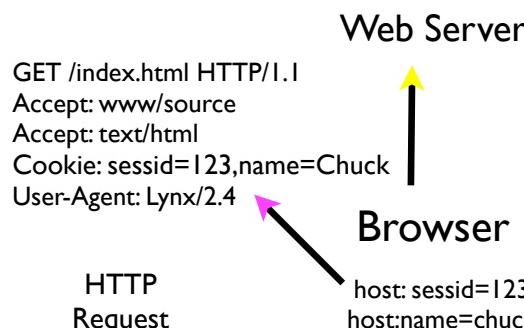
HTTP Request / Response Cycle



On each response, the server can change a cookie value or add another cookie.

http://www.oreilly.com/openbook/cgi/ch04_02.html

HTTP Request / Response Cycle

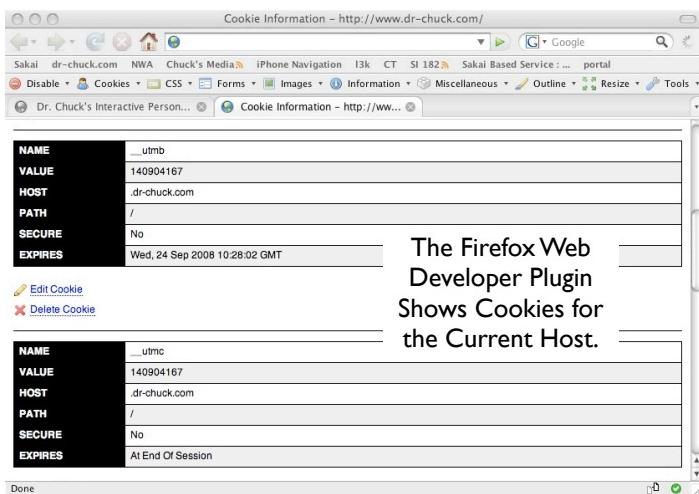


http://www.oreilly.com/openbook/cgi/ch04_02.html

From that point forward, each time we send a GET or POST to the server, we include all the cookies which were set by that host.

Security

- We only send cookies back to the host that originally set the cookie
- The browser has *lots* of cookies for lots of hosts
- To see all Cookies: Firefox -> Preferences -> Privacy -> Show Cookies



Two Kinds of Cookies

- Two kinds of cookie
- Long-lived - who you are - account name last access time - you can close and reopen your browser and it is still there
- Temporary - used to identify your session - it goes away when you close the browser

| | |
|----------------|-------------------------------|
| NAME | _utmb |
| VALUE | 140904167 |
| HOST | .dr-chuck.com |
| PATH | / |
| SECURE | No |
| EXPIRES | Wed, 24 Sep 2008 10:28:02 GMT |

| | |
|----------------|-------------------|
| NAME | _utmz |
| VALUE | 140904167 |
| HOST | dr-chuck.com |
| PATH | / |
| SECURE | No |
| EXPIRES | At End Of Session |

[Edit Cookie](#)
[Delete Cookie](#)

Done

Using Cookies to Support Sessions and Login / Logout

Welcome to SI539

- About
- Contact
- Pictures
- Membership
- Chat
- Application

Please Log In

Required Information

Enter your E-Mail: _____

Enter your Password: _____

Submit

UNIVERSITY OF MICHIGAN WEBLOGIN

AUTHENTICATION REQUIRED:
You are connecting to a U-M website that requires authentication. Please enter your Login ID (username or Friend ID) and password to continue.

Need a Login ID?
If you don't have a Login ID, you can [create one now](#).

By using this service you agree to adhere to [U-M computing policies and guidelines](#).

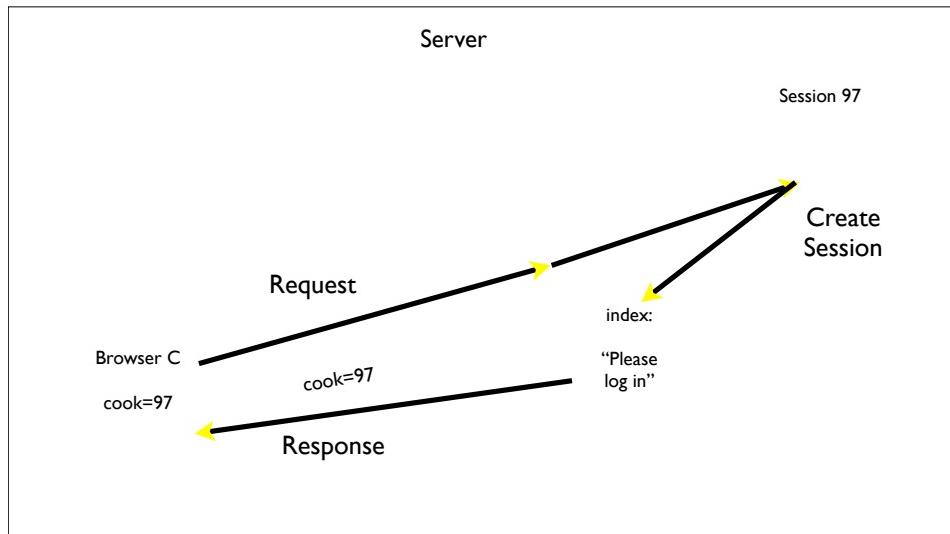
Some Web sites always seem to want to know who you are!

In The Server - Sessions

- In most server applications, as soon as we meet a new browser - we create a session
- We set a session cookie to be stored in the browser which indicates the session id in use
- The creation and destruction of sessions is generally handled by a web framework or some utility code that we just use to manage the sessions

Session Identifier

- A large, random number that we place in a browser cookie the first time we encounter a browser.
- This number is used to pick from the many sessions that the server has active at any one time.
- Server software stores data in the session which it wants to have from one request to another from the same browser.
- Shopping cart or login information is stored in the session in the server



Server

Session 97

Typing

Welcome to SI539

Please Log In

About
Contact
Feedback
Membership
Application

Required Information

Enter your E-Mail:

Enter your Password:

If you have lost your password, please send an E-Mail to membership@SI539.com to have your password reset.

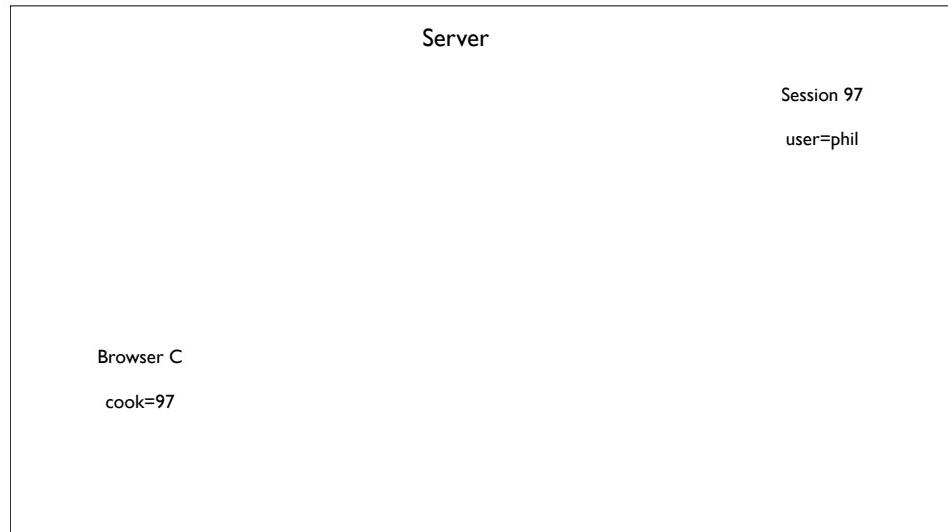
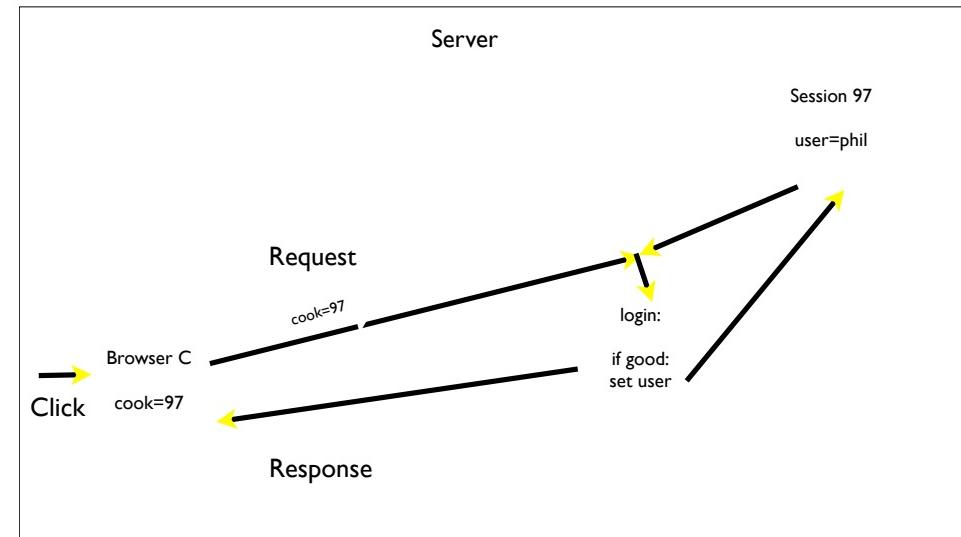
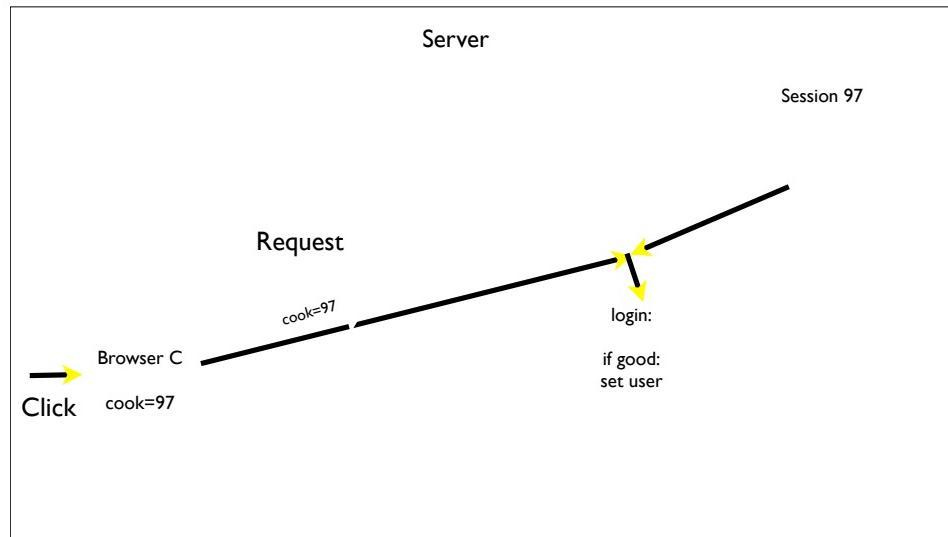
Browser C

cook=97

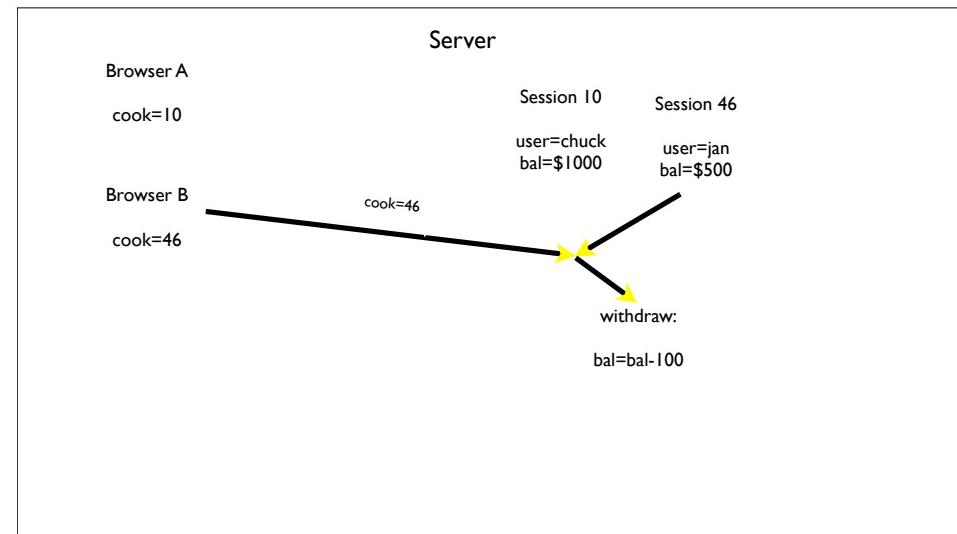
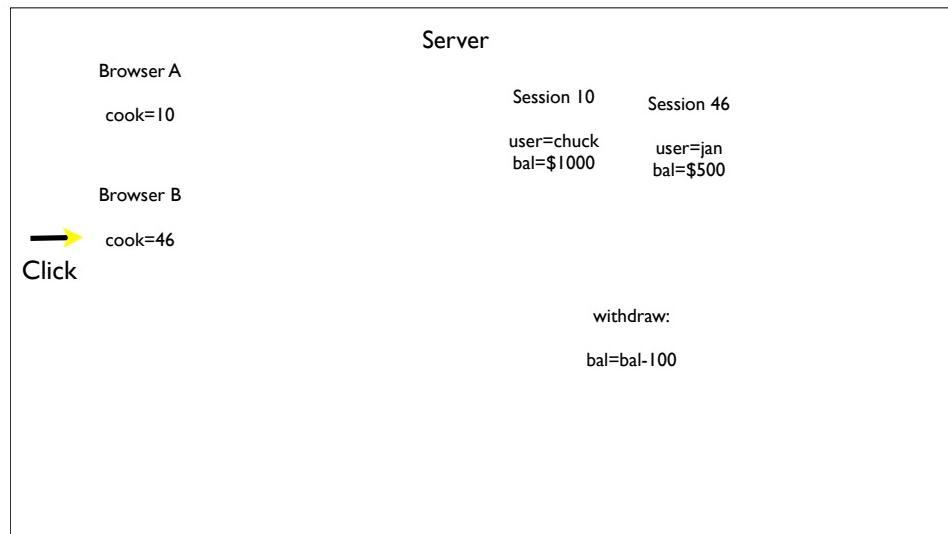
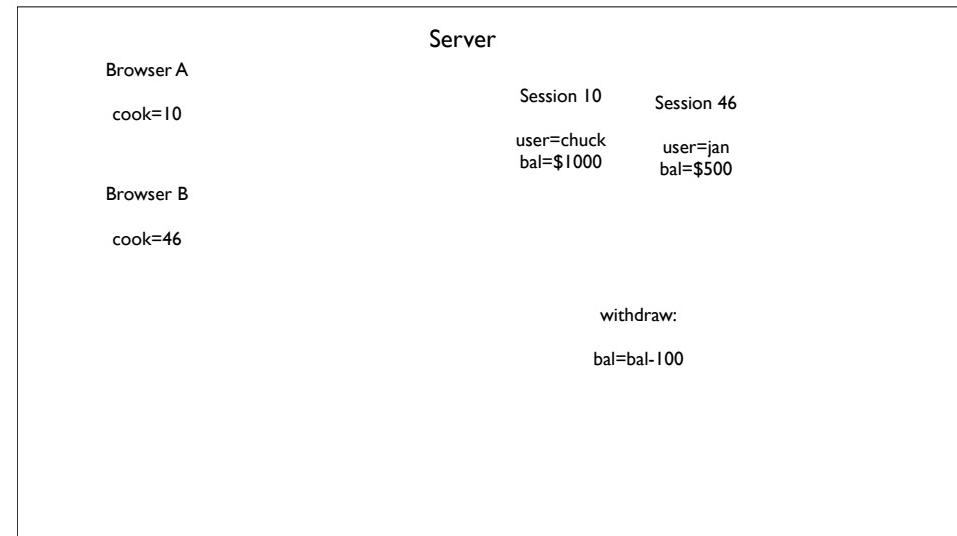
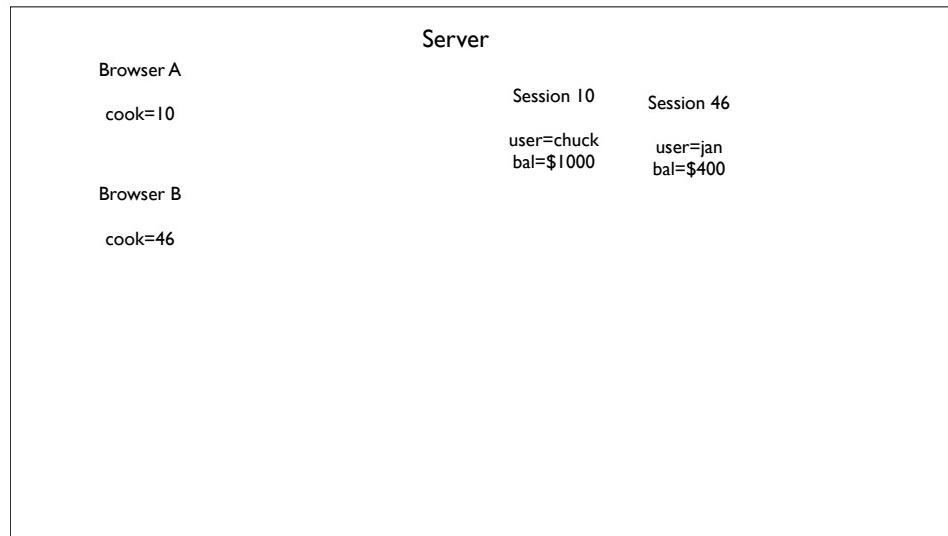
We now have a session established but are not yet logged in.

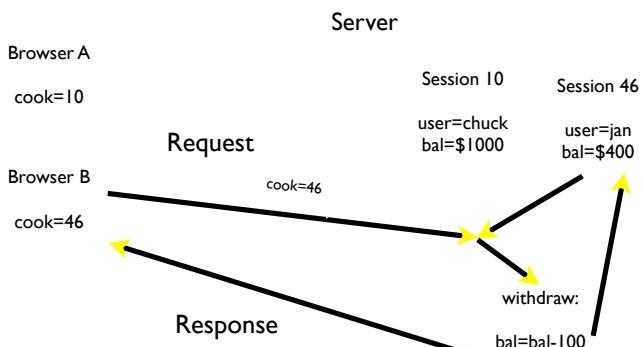
Login / Logout

- Having a session is not the same as being logged in.
- Generally you have a session the instant you connect to a web site
- The Session ID cookie is set when the first page is delivered
- Login puts user information in the session (stored in the server)
- Logout removes user information from the session



Using Sessions for Other Stuff

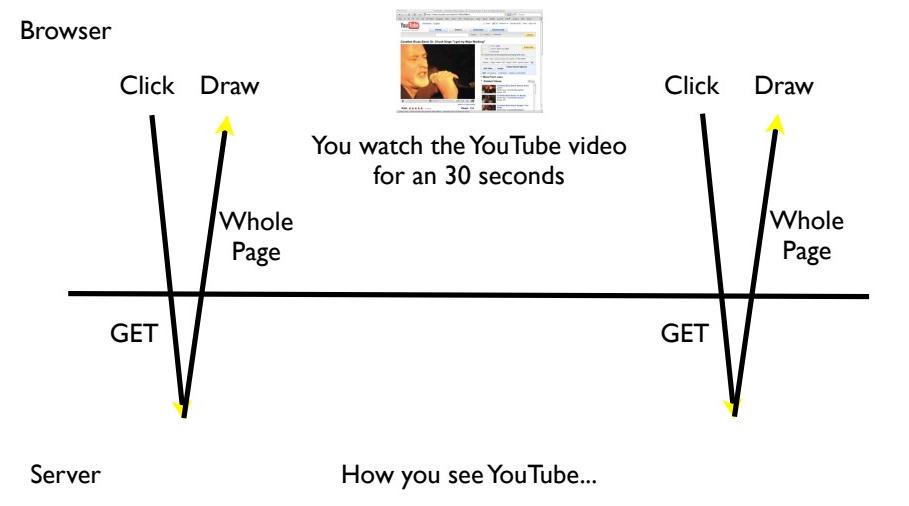


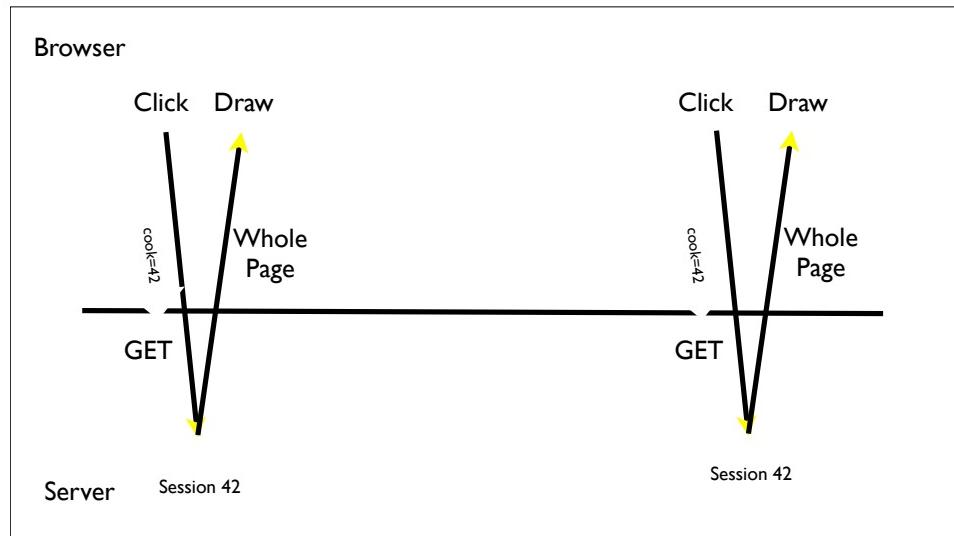
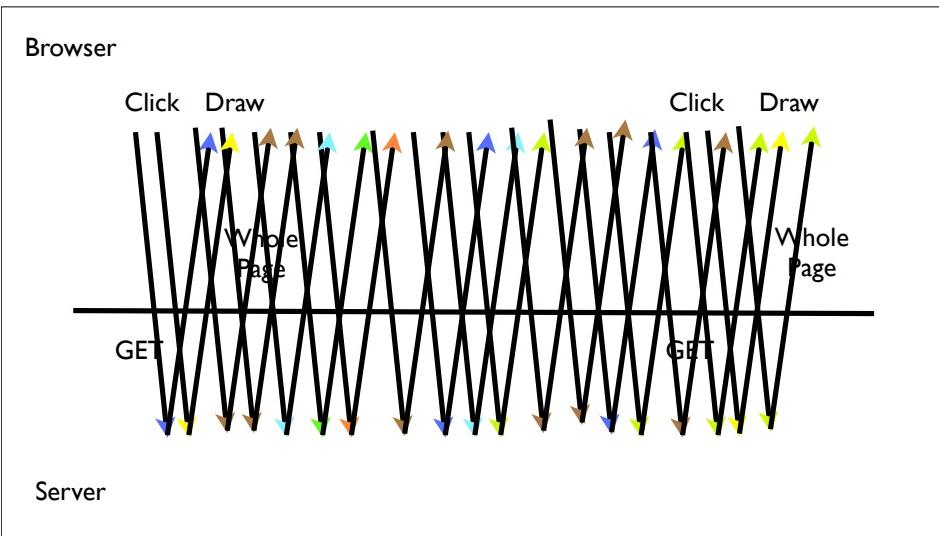


Review...

High Level Summary

- The web is “stateless” - the browser does not maintain a connection to the server while you are looking at a page. You may never come back to the same server - or it may be a long time - or it may be one second later
- So we need a way for servers to know “which browser is this?”
 - In the browser state is stored in “Cookies”
 - In the server state is stored in “Sessions”





Cookie/Session Summary

- Cookies take the stateless web and allow servers to store small “breadcrumbs” in each browser.
- Session IDs are large random numbers stored in a cookie and used to maintain a session on the server for each of the browsers connecting to the server
- Server software stores sessions *somewhere* - each time a request comes back in, the right session is retrieved based on the cookie
- Server uses the session as a scratch space for little things